NOLTR 71-2

FNOL3, A COMPUTER PROGRAM TO SOLVE ORDINARY DIFFERENTIAL EQUATIONS

By
Ralph E. Ferguson
Theodore A. Orlow

1 MARCH 1971

NOL

NAVAL ORDNANCE LABORATORY, WHITE OAK, SILVER SPRING, MARYLAND

ATTENTION

NOLTR 71-2

FNOL3, A COMPUTER PROGRAM TO SOLVE ORDINARY DIFFERENTIAL EQUATIONS

Prepared by:
Ralph E. Ferguson and Theodore A. Orlow

ABSTRACT: FNOL3 is a Fortran IV subprogram which uses fourth order Runge-Kutta and Adams-Moulton methods to solve up to 30 ordinary differential equations with initial conditions. Options allow control of error, step-size, print frequency and integration method. There is a discussion on the integration methods and their error terms. Listings of the subprogram and sample problems and their results as run on NOL's CDC 6400 are included.

U.S. NAVAL ORDNANCE LABORATORY
WHITE OAK, MARYLAND

NOLTR 71-2 1 March 1971

FNOL3, A Computer Program to Solve Ordinary Differential Equations

This report describes a computer program to solve ordinary differential equations which may originate in any of the engineering sciences. It is part of a library of programs available to all users of NOL's CDC 6400 computer.

GEORGE G. BALL
Captain, USN
Commander

E. K. RITTER
By direction

CONTENTS

TABLES

REFERENCES

(a) Linnekin, J.S. and Belliveau, L.J., FNOL2, A FORTRAN (IBM 7090) Subroutine for the Solution of Ordinary Differential Equations with Automatic Adjustment of the Interval of Integration, NOLTR 63-171, 17 Jul 1963

(b) Hildebrand, F.B., Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc., 1956

FNOL3, A COMPUTER PROGRAM TO SOLVE ORDINARY DIFFERENTIAL EQUATIONS

I. INTRODUCTION

This report describes a computer program FNOL3 for the numerical integration of ordinary differential equations with initial conditions. These equations are reduced by the user to a system of first order, simultaneous, ordinary differential equations with initial conditions and are solved by using fourth order Runge-Kutta and fourth order Adams-Moulton predictor-corrector methods. FNOL3 is the successor to the Naval Ordnance Laboratory ordinary differential equation solver FNOL2 and uses many features familiar to users of FNOL2 (the differences are provided in Appendix E). See reference (a).

FNOL3 is formulated so as to be quite separate from any particular application. Options make available a convenient flexible package that can be used whenever the problem is expressible as a system of equations as described above.

The program is written in the FORTRAN IV language for the operating system currently used on the Laboratory's CDC 6400 computer.

II. DESCRIPTION OF THE METHODS

The discussions that follow in this section and in Section III are taken from reference (b). The notation used omits commas from subscripts (f_{in}) except when there might be an ambiguity ($f_{i,n+1}$). Y is used for true values of the dependent variable and y for computed values.

Let the system of equations to be solved be given in the form:

$$y'_i = \frac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_N) \qquad i = 1,2,\ldots,N \qquad (1a)$$

$$y_i(x_0) = y_{i0} \qquad i = 1,2,\ldots,N \qquad (1b)$$

$$T = T(x, y_1, y_2, \ldots, y_N, y'_1, y'_2, \ldots, y'_N) = 0 \qquad (1c)$$

where T is the termination condition and $y_{10}, y_{20}, \ldots, y_{N0}$ are the initial conditions at $x = x_0$.

Let y_{in} be the value of y_i and f_{in} be the value of f_i at $x=x_n$. Let h be the step size in the independent variable x. The Runge-Kutta method uses the following formulae with the appropriate initial conditions to go from step n to step n+1.

$$k_{i1} = f_i(x_n, y_{1n}, \ldots, y_{Nn}) \qquad i = 1,2,\ldots,N \qquad (2a)$$

$$k_{i2} = f_i(x_n+\tfrac{1}{2}h, y_{1n}+\tfrac{1}{2}hk_{11}, \ldots, y_{Nn}+\tfrac{1}{2}hk_{N1}) \qquad i = 1,2,\ldots,N \qquad (2b)$$

$$k_{i3} = f_i(x_n+\tfrac{1}{2}h, y_{1n}+\tfrac{1}{2}hk_{12}, \ldots, y_{Nn}+\tfrac{1}{2}hk_{N2}) \qquad i = 1,2,\ldots,N \qquad (2c)$$

$$k_{i4} = f_i(x_n+h, y_{1n}+hk_{13}, \ldots, y_{Nn}+hk_{N3}) \qquad i = 1,2,\ldots,N \qquad (2d)$$

$$y_{i,n+1} = y_{in}+\frac{h}{6}(k_{i1}+2k_{i2}+2k_{i3}+k_{i4}) \qquad i = 1,2,\ldots,N \qquad (2e)$$

The Adams-Moulton predictor-corrector method uses the following formulae to compute the values of $y_{i,n+1}$ using the values of $y_{i,n-3}$, $y_{i,n-2}$, $y_{i,n-1}$, and y_{in}.

$$y^{(p)}_{i,n+1} = y_{in}+\frac{h}{24}(55f_{in}-59f_{i,n-1}+37f_{i,n-2}-9f_{i,n-3}) \qquad i = 1,2,\ldots,N \qquad (3a)$$

$$y^{(c)}_{i,n+1} = y_{in}+\frac{h}{24}(9f^{(p)}_{i,n+1}+19f_{in}-5f_{i,n-1}+f_{i,n-2}) \qquad i = 1,2,\ldots,N \qquad (3b)$$

where the values of y_i at x_1, x_2, and x_3 are found by using the Runge-Kutta formulae. x_4 is the first point at which the Adams-Moulton method is used.

Observe that the Adams-Moulton formulae only require two derivative evaluations to go to the next step. The Runge-Kutta method requires four derivative evaluations for each step.

III. THE ERROR TERMS

We will assume in this section only one ordinary differential equation, so the i from y_{in} and f_{in} is omitted. The error term for the Adams-Moulton method is as follows:

$$y_{n+1}^{(p)} = y_n+\frac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}) \tag{1a}$$

$$y_{n+1}^{(c)} = y_n+\frac{h}{24}(9f_{n+1}^{(p)}+19f_n-5f_{n-1}+f_{n-2}) \tag{1b}$$

where p stands for predicted value and c for corrected value. $f_{n+1}^{(p)}$ means calculate f at x_{n+1} using $y_{n+1}^{(p)}$.

If the calculated values of $y_1, y_2, \ldots, y_n$ and accordingly, of $f_1, f_2, \ldots, f_n$ were exactly correct, then the true ordinate at x_{n+1}, say Y_{n+1} would satisfy the equations

$$Y_{n+1} = y_n+\frac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3})+\frac{251}{720}h^5 y^V(\xi_1) \tag{2a}$$

$$Y_{n+1} = y_n+\frac{h}{24}(9f_{n+1}+19f_n-5f_{n-1}+f_{n-2}) - \frac{19}{720}h^5 y^V(\xi_2) \tag{2b}$$

where ξ_1 and ξ_2 both lie between x_{n-3} and x_{n+1}. $y^V(\xi_1)$ is the fifth derivative of y at $x = \xi_1$. So it follows from (1) and (2) that

$$Y_{n+1}-y_{n+1}^{(p)} = \frac{251}{720} h^5 y^V(\xi_1) \tag{3a}$$

$$Y_{n+1}-y_{n+1}^{(c)} = \frac{3h}{8}(f_{n+1}-f_{n+1}^{(p)}) - \frac{19}{720}h^5 y^V(\xi_2) \tag{3b}$$

Let $F(x_{n+1},Y_{n+1}) = f_{n+1}(x_{n+1},Y_{n+1})$, applying the law of the mean

$$f_{n+1} - f_{n+1}^{(p)} = F(x_{n+1},Y_{n+1}) - F(x_{n+1},y_{n+1}^{(p)}) = (Y_{n+1}-y_{n+1}^{(p)})F_y(x_{n+1},\eta_{n+1}) \tag{4}$$

where η_{n+1} is between Y_{n+1} and $y_{n+1}^{(p)}$.

It is assumed that h is sufficiently small to ensure that

$$\frac{3h}{8}\left|F_y(x_{n+1},\eta_{n+1})\right| \ll 1 \tag{5}$$

and also that $y^V(x)$ does not vary strongly for $x_{n-3} < x < x_{n+1}$ so that $y^V(\xi_1)$ and $y^V(\xi_2)$ can be equated. Equations (3) lead to the useful approximate relation

$$\frac{720}{251}(Y_{n+1}-y_{n+1}^{(p)}) = h^5 y^V(\xi)$$

$$Y_{n+1} - y_{n+1}^{(c)} = \frac{3h}{8}(Y_{n+1}-y_{n+1}^{(p)}) F_y(x_{n+1},\eta_{n+1})-(\frac{19}{720})(\frac{720}{251})(Y_{n+1}-y_{n+1}^{(p)})$$

$$Y_{n+1}-y_{n+1}^{(c)} = (\frac{3h}{8} F_y(x_{n+1},\eta_{n+1}) - \frac{19}{251})(Y_{n+1}-y_{n+1}^{(p)})$$

so from (5)

$$Y_{n+1} - y^{(c)}_{n+1} \cong -\frac{19}{251}(Y_{n+1}-y^{(p)}_{n+1})$$

$$Y_{n+1} + \frac{19}{251} Y_{n+1} \cong y^{(c)}_{n+1} + \frac{19}{251} y^{(p)}_{n+1}$$

$$\frac{270}{251} Y_{n+1} \cong \frac{270}{251} y^{(c)}_{n+1} - \frac{19}{251} y^{(c)}_{n+1} + \frac{19}{251} y^{(p)}_{n+1}$$

Therefore,

$$Y_{n+1} \cong y^{(c)}_{n+1} + \frac{1}{14} (y^{(p)}_{n+1} - y^{(c)}_{n+1}) \tag{6}$$

This is the equation programmed when the Adams-Moulton method is selected.

The fourth order Runge-Kutta formulae are

$$k_1 = f(x_n,y_n) \tag{7a}$$

$$k_2 = f(x_n+\tfrac{1}{2}h,y_n+\tfrac{1}{2}hk_1) \tag{7b}$$

$$k_3 = f(x_n+\tfrac{1}{2}h,y_n+\tfrac{1}{2}hk_2) \tag{7c}$$

$$k_4 = f(x_n+h,y_n+hk_3) \tag{7d}$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1+2k_2+2k_3+k_4). \tag{7e}$$

No simple expressions are known for the precise truncation errors in the preceding formulae. An estimate of the error can be obtained, in practice, in the following way. Let the truncation error associated with a formula of 4th-order accuracy in advancing from x_n to that at $x_{n+1} = x_n + h$ in a single step be denoted by $c_n\, h^5$. Also suppose that c_n varies "slowly" with n and is nearly independent of h when h is small. Then if the true value of y at x_{n+1} is denoted by Y_{n+1}, the value obtained by two steps starting at x_{n-1} by $y^{(h)}_{n+1}$, and the value obtained by a single step with doubled spacing 2h by $y^{(2h)}_{n+1}$, there follows approximately

$$Y_{n+1} - y_{n+1}^{(h)} \cong 2c_n h^5 \tag{8a}$$

$$Y_{n+1} - y_{n+1}^{(2h)} \cong c_n (2h)^5 \tag{8b}$$

when h is small.

From equation (8b)

$$Y_{n+1} - y_{n+1}^{(2h)} \cong 2^5 c_n h^5$$

so

$$Y_{n+1} - y_{n+1}^{(2h)} \cong 2^4 (Y_{n+1} - y_{n+1}^{(h)})$$

Therefore,

$$Y_{n+1} \cong y_{n+1}^{(h)} + \frac{y_{n+1}^{(h)} - y_{n+1}^{(2h)}}{15} \tag{9}$$

This is the equation programmed when the Runge-Kutta method is selected.

IV. AUTOMATIC ADJUSTMENT OF STEP SIZE

FNOL3 has the option of automatically varying the step size h, to hold the truncation errors within bounds fixed by the user. The absolute truncation errors for the Adams-Moulton method from III-6 are

$$AE_i = \frac{y_{i,n+1}^{(p)} - y_{i,n+1}^{(c)}}{14} \qquad i = 1,2,\ldots,N \tag{1}$$

and the relative truncation errors are

$$RE_i = \frac{AE_i}{y_{i,n+1}^{(c)}} \qquad i = 1,2,\ldots,N \tag{2}$$

The absolute truncation errors for the Runge-Kutta method from III-9 are

$$AE_i = \frac{y_{i,n+1}^{(h)} - y_{i,n+1}^{(2h)}}{15} \qquad i = 1,2,\ldots,N \tag{3}$$

and the relative truncation errors are

$$RE_i = \frac{AE_i}{y_{i,n+1}^{(h)}} \qquad i = 1,2,\ldots,N \tag{4}$$

To determine the step size at each step, let $E_1, E_2, \ldots, E_N$ be the errors whether they are absolute or relative errors and let XNE be a step size control parameter (see page 8). Then if $10^{-XNE-3} \leq |E_i| \leq 10^{-XNE}$ for $i = 1,2,\ldots,N$, the step size is not changed. But if $|E_i| \leq 10^{-XNE-3}$ for all i, then let

$$HB = \underset{i}{\text{minimum}} \frac{10^{-XNE-1.5}}{|E_i|+10^{-11}} > 1 \tag{5}$$

and the step size is increased to $HB^{1/5} \cdot h$. If $|E_j| > 10^{-XNE}$ for some j then

$$HB = \underset{j}{\text{minimum}} \frac{10^{-XNE-1.5}}{|E_j|+10^{-11}} < 1 \tag{6}$$

and the step size is decreased to $HB^{1/5} \cdot h$.

V. PROGRAMMING

The user must write a calling program hereafter called MAIN, and three auxiliary subprograms. The latter are usually called DERIV, TERM, and OUTPUT and are described later in this section. Besides calling on FNOL3 and providing it with initial values, MAIN must contain an EXTERNAL card which denotes that DERIV, TERM and OUTPUT are the names of subprograms, not variables.

The calling sequence for FNOL3 is

```
EXTERNAL DERIV, TERM, OUTPUT
CALL FNOL3 (J,N,G,L,M,XNE,X,Y,D,DERIV,TERM,OUTPUT)
```

J: (INPUT,INTEGER)

This parameter indicates the integration method.

J = 1 Use Runge-Kutta method of integration to termination. Truncation errors are not calculated; the step size G is not adjustable.

J = 2 Use Runge-Kutta for the first three steps, then Adams-Moulton for the remainder of the interval of integration. Truncation errors are calculated. The step size is adjustable unless XNE = 0. If the step size is adjusted, new starting values are obtained through the Runge-Kutta method.

J = 3 Use Runge-Kutta throughout. The truncation errors are calculated; the step size is adjustable unless XNE = 0.

N: (INPUT,INTEGER)

This is the number of simultaneous first order differential equations to be solved. The maximum number of equations is 30.

G: (INPUT,REAL)

This is the initial step size; upon return from FNOL3, G retains its original value.

L: (INPUT,INTEGER)

This is the number of Y's beyond (N+3) to be written in the routine OUTPUT. These additional Y's should be calculated in the routine TERM, beginning with Y(N+3+1) to Y(N+3+L).

M: (INPUT,INTEGER)

This is the number of accepted steps taken between calls to routine OUTPUT. If M = 0, then printing is determined by values assigned to Y(N+1) and Y(N+2). Y(N+1) = f(X,Y(1),...,Y(N),D(1),...,D(N)) and is defined in routine TERM. To generate a call to OUTPUT, Y(N+1) must change by an amount greater than Y(N+2) since the last call to OUTPUT. Y(N+2) is assigned a constant value in the routine which calls FNOL3.

XNE: (INPUT,REAL)

This is the step size control. The step size is unchanged if the worst of all the errors lies within the window $[10^{-XNE-3}, 10^{-XNE}]$. The step size is increased if the errors are all less than 10^{-XNE-3}. The step size is decreased if for some differential equation the error is greater than 10^{-XNE}.

If Y(N+3) < 0. and XNE ≠ 0., the automatic adjustment of the step size is a function of the absolute errors.

If Y(N+3) = 0. and XNE ≠ 0., the automatic adjustment of the step size is a function of the relative errors.

If Y(N+3) = ε > 0. and XNE ≠ 0., the automatic adjustment of the step size is a function of the relative errors where the relative errors are equal to the absolute errors divided by the maximum (Y(N+3), |Y(I)|). This option removes the possibility of using "small" functional values to compute relative error, otherwise this option is identical to the previous option and is to be preferred over it.

If XNE = 0., the step size G is not adjustable.

X: (INPUT,OUTPUT,REAL)

This is the independent variable. An initial value must be specified before calling FNOL3. FNOL3 will return with X,the terminal value of the independent variable.

Y: (INPUT,OUTPUT,REAL)

This is the name given the solution array. Y must be dimensioned at least Y(N+3+L). Initial values for Y(1),Y(2),...,Y(N) must be specified before calling FNOL3. If L > 0, then Y(N+3+1),...,Y(N+3+L) may be used to calculate additional values in the routine TERM. Upon returning from FNOL3, Y(1),...,Y(N),Y(N+3+1),...,Y(N+3+L) have the values computed at the terminal value of X.

D: (OUTPUT,REAL)

This is the name given to the array where the derivatives are stored and must be defined in routine DERIV. D should be dimensioned D(N). Upon returning from FNOL3, D(1),...,D(N) have the values computed at the terminal value of X.

DERIV:

In this routine the user must compute the N derivatives. The general form is

```
SUBROUTINE DERIV (X,Y,D)
DIMENSION Y(1),D(1)
D(1) = ...
D(2) = ...
  .
  .
  .
D(N) = ...
RETURN
END
```

No D beyond D(N) should be calculated. If desired, additional data may be passed between this routine and the other user written routines via COMMON statements.

TERM:

The user evaluates the termination criterion (T = T(X,Y(1),...,Y(N), D(1),...,D(N)) in this routine. Auxiliary values such as Y(N+1) and those required for plotting purposes should be calculated here because this routine is entered only once per accepted step before termination.

A termination loop of at most four iterations starts when T undergoes a change of sign. After the very first step of the integration, termination may occur without looping when $|T| \le 10^{-6}$.

The general form is

```
SUBROUTINE TERM (X,Y,D,T)
DIMENSION Y(1),D(1)
T = ···
RETURN
END
```

OUTPUT:

This user routine is entered at the beginning and end of the complete integration. If M ≠ 0, it is entered every Mth accepted step. If M = 0 then Y(N+1) and Y(N+2), discussed under parameter M determine the print frequency.

The general form is

```
SUBROUTINE OUTPUT (X,Y,D,ERROR,N,L,H)
DIMENSION Y(1),D(1),ERROR(1)
    :
PRINT ···
    :
RETURN
END
```

X,Y,D,N, and L are the same as in the calling sequence of FNOL3.
ERROR is the name given the array which contains the absolute errors.

H is the step size used to get to the present step. If the magnitude of a new step size becomes less than $H_1 = 10^{-6} * |x| + 10^{-11}$, then two steps are taken using the Runge-Kutta method with step size equal to H_1 or $-H_1$ depending on the direction of integration. If the errors are too large, the two steps will be accepted and two more steps are taken using the Runge-Kutta method with a step size computed in the same way as above using the current x. If the errors are still too large after the above procedure is done ten times, then an error message is written and the program stops after going to OUTPUT. If at any step the errors are not too large, then FNOL3 continues in the normal manner after resetting the counter back to zero.

VI. EXAMPLES

The examples section of this report contains three problems which illustrate the available options of FNOL3.

Example No. 1

Solve the first order linear differential equation

$$\frac{dy}{dx} = \omega A \cos(\omega x)$$

in the interval $[0,3\pi]$ for the initial condition

$$y(0) = 0$$

Choose the case $\omega = A = 1$.

Use the three available methods of FNOL3 to numerically integrate the given differential equation over $[0,3\pi]$.

(I) Use the fourth order Runge-Kutta method with a constant step size of .1.

(II) Use the fourth order Adams-Moulton predictor-corrector formulae with an adjustable step size to hold the relative truncation error in the window $[10^{-8},10^{-5}]$.

(III) Use the fourth order Runge-Kutta method with an adjustable step size to hold the relative truncation error in the window $[10^{-8},10^{-5}]$.

The analytic solution of the given equation with respect to the given initial condition is

$$y(x) = A \sin(\omega x).$$

The actual value of $\sin(x)$ is computed for each x and compared to the calculated value of the solution. At every fifth accepted step, the solution, $\sin(x)$, and the difference of the actual and calculated solutions are printed. These results are in Table 1.

The main (calling) program and the necessary subroutines to accomplish the solution of the stated problem in the manner indicated are in Appendix A.

Example No. 2

Solve the second order linear differential equation

$$\frac{d^2y}{dx^2} = -y$$

in the interval $[0,5\pi]$ for the initial conditions

$y(0) = 0$

$y'(0) = 1$

Reduce the second order linear differential equation to a system of two first order equations

$$\frac{dy_1}{dx} = y_2$$

$$\frac{dy_2}{dx} = -y_1$$

with the initial conditions

$y_1(0) = 0$

$y_2(0) = 1$

Use the Adams-Moulton predictor-corrector method to integrate the given differential equations over $[0,5\pi]$. An adjustable step size is used with size determined by

(I) Relative truncation error

(II) Absolute truncation error

The error window is $[10^{-8}, 10^{-5}]$.

Two types of printing options are illustrated in this example

(I) Printing every kth step (10 in this example)

(II) Printing when the independent variable has changed by a pre-selected amount (1 in this example)

The analytic solution of the given equation with respect to the given initial conditions is

$y(x) = \sin(x)$

The actual value of sin(x) is computed for each x and compared to the calculated value of the solution. At every print cycle the solution, sin(x), and the difference of the actual and calculated solutions are printed. These results are in Table 2.

The main (calling) program and the necessary subroutines to accomplish the solution of the stated problem in the manner indicated are in Appendix B.

Example No. 3

Solve the system of first order linear differential equations

$$\frac{dy_1}{dx} = y_2 \tag{1}$$

$$\frac{dy_2}{dx} = -y_1 \tag{2}$$

$$\frac{dy_3}{dx} = \left(\frac{1}{x+1} - 1\right) y_3 \tag{3}$$

$$\frac{dy_4}{dx} = y_4 - x+1 \tag{4}$$

$$\frac{dy_5}{dx} = \frac{1}{x+1} \tag{5}$$

in the interval $[0,5\pi]$ for the initial conditions

$$y_1(0) = 0$$

$$y_2(0) = 1$$

$$y_3(0) = 1$$

$$y_4(0) = 1$$

$$y_5(0) = 0$$

Use the Adams-Moulton predictor-corrector method to integrate the given system of differential equations over $[0,5\pi]$. An adjustable step size is used with size adjusted to hold the relative truncation error in the window $[10^{-8},10^{-5}]$.

Printing occurs every 40 accepted steps. The analytic solutions of the given equations with respect to the given initial conditions are

$$y_1(x) = \sin(x)$$

$$y_2(x) = \cos(x)$$

$$y_3(x) = (x+1)e^{-x}$$

$$y_4(x) = x+e^{x}$$

$$y_5(x) = \log_e(x+1)$$

The actual values of the solutions are computed at each x and compared to the calculated values of the solutions. At each print cycle the actual solutions, calculated solutions, and differences are printed. These results are in Table 3.

The main (calling) program and the necessary subroutines to accomplish the solution of the stated problem in the manner indicated are in Appendix C.

VII. REMARKS

Some suggestions and warnings are in order based on the years of experience with previous versions of this routine. This version of FNOL3 is a conversion from the IBM 7090, and what was one auxiliary routine with multiple entries has been changed to three auxiliary routines. In addition, double precision has been deemed unnecessary due to the increased precision of the CDC 6400.

J: Adams-Moulton is the prime method of this routine, but Runge-Kutta is needed to start, and whenever the time step changes. Adams-Moulton requires only 2 entries into DERIV per time step while Runge-Kutta requires 4.

L: This parameter is a hangover from previous versions of FNOL3 in which the user did not write an OUTPUT routine but simply specified how many (L) auxiliary values were to be printed.

M: FNOL3 always prints initial and final values even if M has been set to some very large value. However, some output is always needed to shed light on how the integration has proceeded.

Printing at specified intervals cannot be insured except by fixed timesteps or terminating and then printing. Setting M = 0 and Y(N+2)=DELTAY in MAIN and Y(N+1)=Y(1) in TERM, only causes no printing until Y(1) changes by at least Y(N+2). To force printing at exact intervals of Y(N+2) in Y(1), put FNOL3 in a loop in MAIN and in TERM set T=Y(1)-C. In MAIN set C=Y(1)+Y(N+2) before calling FNOL3. C may be transmitted via the Y array or through COMMON. The final value of Y(1) of each integration becomes the initial value for the next integration.

Y: Unless COMMON is used, additional parameters necessary for computation should be sent from routine to routine through the Y array in locations starting with Y(N+4). If M $\neq$ 0 then Y(N+1) and Y(N+2) are available to the programmer. If XNE = 0 then Y(N+3) is available. In other cases these three locations have special meaning. See the discussions on M and XNE above and in Section V.

D: We repeat here for emphasis. DO NOT USE a D beyond D(N)! In all we have said, there is an assumption that the derivatives can be computed explicitly from X and the current Y values. If this is not so, the alternative is to perform root-finding within DERIV. If the routine that calls FNOL3 initializes the D array, these values will be available in DERIV. However, unless they are saved in another array, each time DERIV is entered, the previous D's are lost.

ERROR: At the start of an integration or whenever the step size changes, the absolute truncation errors are not computed directly for the first step. Runge-Kutta is called at that time, and its error term requires having $y_{i,n+1}^{(h)}$ and $y_{i,n+1}^{(2h)}$, see equation IV-3. The first term, $y_{i,n+1}^{(h)}$, means reaching x_{n+1} by taking two steps of size h, while the second term, $y_{i,n+1}^{(2h)}$, means reaching x_{n+1} by taking one step of size 2h. It is only after these three steps are taken that an error term can be computed. Accepting or rejecting the first step is based on this error, which is then treated as if it were the first error. At least three Runge-Kutta steps of size h are taken before Adams-Moulton takes over.

VIII. ACKNOWLEDGMENT

NOL has had an ordinary differential equations solver since 1959, and FNOL3 is the current version. We offer our sincere thanks to L. Hieb, M. Vander Vorst and J. Lyles who helped to checkout and criticise the FNOL3 program and writeup. A number of improvements were made on the basis of their suggestions. Mrs. G. Dabbs then did a commendable job of typing the document.

TABLE 1

$$\frac{dy}{dx} = \omega A \cos(\omega x)$$

Method (I)

x	YT=TRUE sin(x)	YC=COMPUTED sin(x)	(YT-YC)
0.0	0.0	0.0	0.0
0.5	.479426	.479426	$-17x10^{-9}$
1.0	.841471	.841471	$-29x10^{-9}$
1.5	.997495	.997495	$-35x10^{-9}$
2.0	.909297	.909297	$-32x10^{-9}$
2.5	.598472	.598472	$-21x10^{-9}$
3.0	.141120	.141120	$-49x10^{-10}$
3.5	-.350783	-.350783	$12x10^{-9}$
4.0	-.756802	-.756803	$26x10^{-9}$
4.5	-.977530	-.977530	$34x10^{-9}$
5.0	-.958924	-.958924	$33x10^{-9}$
5.5	-.705540	-.705540	$25x10^{-9}$
6.0	-.279415	-.279416	$97x10^{-10}$
6.5	.215120	.215120	$-75x10^{-10}$
7.0	.656987	.656987	$-23x10^{-9}$
7.5	.938000	.938000	$-33x10^{-9}$
8.0	.989358	.989358	$-34x10^{-9}$
8.5	.798487	.798487	$-28x10^{-9}$
9.0	.412118	.412118	$-14x10^{-9}$
9.424778	$-39x10^{-9}$	$-41x10^{-9}$	$17x10^{-10}$

TABLE 1

$$\frac{dy}{dx} = \omega A \cos(\omega x)$$

Method (II)

x	YT=TRUE sin(x)	YC=COMPUTED sin(x)	(YT-YC)
0.0	0.0	0.0	0.0
0.5	.479426	.479426	$22\text{x}10^{-9}$
1.0	.841471	.841471	$92\text{x}10^{-9}$
1.5	.997495	.997495	$18\text{x}10^{-8}$
2.216974	.798391	.798391	$24\text{x}10^{-8}$
2.822212	.313979	.313979	$16\text{x}10^{-8}$
3.219365	-.0776944	-.0776946	$17\text{x}10^{-8}$
3.729709	-.554795	-.554795	$13\text{x}10^{-8}$
4.240053	-.890508	-.890508	$47\text{x}10^{-9}$
4.750397	-.999278	-.999278	$-59\text{x}10^{-9}$
5.260741	-.853385	-.853384	$-16\text{x}10^{-8}$
5.771085	-.490009	-.490009	$-23\text{x}10^{-8}$
6.220706	-.0624391	-.0624388	$-25\text{x}10^{-8}$
6.658950	.366984	.366985	$-24\text{x}10^{-8}$
7.155075	.765546	.765546	$-18\text{x}10^{-8}$
7.651199	.979510	.979510	$-92\text{x}10^{-9}$
8.353455	.877835	.877835	$29\text{x}10^{-8}$
9.288410	.135945	.135946	$-61\text{x}10^{-8}$
9.424778	$-39\text{x}10^{-9}$	$62\text{x}10^{-8}$	$-66\text{x}10^{-8}$

TABLE 1

$$\frac{dy}{dx} = \omega A \cos(\omega x)$$

Method (III)

x	YT=TRUE sin(x)	YC=COMPUTED sin(x)	(YT-YC)
0.0	0.0	0.0	0.0
0.5	.479426	.479426	$12\text{x}10^{-9}$
1.303726	.964548	.964548	$91\text{x}10^{-9}$
2.309936	.739048	.739048	$77\text{x}10^{-9}$
3.316146	-.173668	-.173668	$-46\text{x}10^{-8}$
4.322356	-.924896	-.924895	$-10\text{x}10^{-7}$
5.297378	-.833718	-.833717	$-11\text{x}10^{-7}$
6.225618	-.0575354	-.0575345	$-83\text{x}10^{-8}$
7.153858	.764763	.764763	$-41\text{x}10^{-8}$
8.082099	.974094	.974094	$-23\text{x}10^{-8}$
9.102580	.316653	.316653	$-51\text{x}10^{-8}$
9.424778	$-39\text{x}10^{-9}$	$74\text{x}10^{-8}$	$-78\text{x}10^{-8}$

TABLE 2

$$\frac{d^2y}{dx^2} = -y$$

Relative Truncation Error

x	YC=COMPUTED sin(x)	YT=TRUE sin(x)	(YT-YC)
0.0	0.0	0.0	0.0
0.381345	.372169	.372169	$27\text{x}10^{-9}$
1.194792	.930140	.930139	$-22\text{x}10^{-8}$
2.008240	.905838	.905838	$-52\text{x}10^{-8}$
2.821688	.314477	.314476	$-39\text{x}10^{-8}$
3.322058	-.179488	-.179488	$11\text{x}10^{-9}$
4.055814	-.792088	-.792087	$72\text{x}10^{-8}$
4.710873	-1.00000	-.999999	$12\text{x}10^{-7}$
5.262094	-.852680	-.852679	$11\text{x}10^{-7}$
5.953946	-.323324	-.323324	$61\text{x}10^{-8}$
6.645797	.354718	.354717	$-30\text{x}10^{-8}$
7.337648	.869636	.869635	$-12\text{x}10^{-7}$
8.029500	.984638	.984636	$-17\text{x}10^{-7}$
8.721351	.646836	.646835	$-13\text{x}10^{-7}$
9.413203	.0115755	.0115752	$-32\text{x}10^{-8}$
10.105054	-.629009	-.629008	$99\text{x}10^{-8}$
10.796905	-.980332	-.980330	$20\text{x}10^{-7}$
11.488757	-.880833	-.880831	$21\text{x}10^{-7}$
12.180608	-.376267	-.376266	$12\text{x}10^{-7}$
12.872459	.301332	.301332	$-34\text{x}10^{-8}$
13.564311	.840358	.840356	$-19\text{x}10^{-7}$
14.256162	.992931	.992928	$-27\text{x}10^{-7}$
14.948014	.688887	.688885	$-22\text{x}10^{-7}$
15.639865	.0680464	.0680457	$-69\text{x}10^{-8}$
15.707963	$76\text{x}10^{-8}$	$27\text{x}10^{-8}$	$-50\text{x}10^{-8}$

TABLE 2

$$\frac{d^2y}{dx^2} = -y$$

Absolute Truncation Error

x	YC=COMPUTED sin(x)	YT=TRUE sin(x)	(YT-YC)
0.0	0.0	0.0	0.0
1.008358	.845958	.845957	-91×10^{-8}
2.030260	.896293	.896290	-30×10^{-7}
3.052163	.0893123	.0893108	-16×10^{-7}
4.074065	-.803100	-.803096	46×10^{-7}
5.095968	-.927340	-.927331	89×10^{-7}
6.117870	-.164568	-.164563	43×10^{-7}
7.139773	.755619	.755612	-75×10^{-7}
8.161675	.953050	.953035	-15×10^{-6}
9.183578	.238876	.238868	-79×10^{-7}
10.205480	-.703788	-.703778	96×10^{-7}
11.227383	-.973273	-.973253	21×10^{-6}
12.249285	-.311811	-.311799	12×10^{-6}
13.271187	.647905	.647894	-11×10^{-6}
14.293090	.987895	.987869	-27×10^{-6}
15.314992	.382952	.382935	-17×10^{-6}
15.707963	76×10^{-7}	27×10^{-8}	-74×10^{-7}

TABLE 3

x	EQUATION NO.	YC=COMPUTED y	YT=TRUE y	(YT-YC)
0.0	1	0.0	0.0	0.0
	2	1.0	1.0	0.0
	3	1.0	1.0	0.0
	4	1.0	1.0	0.0
	5	0.0	0.0	0.0
2.3055	1	.742059	.742058	-13×10^{-8}
	2	-.670335	-.670335	91×10^{-9}
	3	.329598	.329598	-20×10^{-9}
	4	12.3342	12.3342	15×10^{-7}
	5	1.19557	1.19557	19×10^{-8}
4.6901	1	-.999753	-.999753	34×10^{-8}
	2	-.0222392	-.0222391	63×10^{-9}
	3	-.0522659	-.0522659	-37×10^{-10}
	4	113.559	113.559	35×10^{-6}
	5	1.73874	1.73874	20×10^{-8}
7.0748	1	.711517	.711516	-31×10^{-8}
	2	.702670	.702669	-43×10^{-8}
	3	.00683235	.00683235	-18×10^{-11}
	4	1188.93	1188.93	59×10^{-5}
	5	2.08875	2.08875	20×10^{-8}

TABLE 3

x	EQUATION NO.	YC=COMPUTED y	YT=TRUE y	(YT-YC)
9.4595	1	-.0347456	-.0347457	$-98\text{x}10^{-9}$
	2	-.999397	-.999396	$71\text{x}10^{-8}$
	3	$.815249\text{x}10^{-3}$	$.815249\text{x}10^{-3}$	$43\text{x}10^{-12}$
	4	12839.3	12839.3	$86\text{x}10^{-4}$
	5	2.34751	2.34751	$20\text{x}10^{-8}$
11.8442	1	-.660999	-.660999	$70\text{x}10^{-8}$
	2	.750388	.750387	$-56\text{x}10^{-8}$
	3	$.922205\text{x}10^{-4}$	$.922205\text{x}10^{-4}$	$14\text{x}10^{-12}$
	4	139289.	139289.	$12\text{x}10^{-2}$
	5	2.55289	2.55289	$20\text{x}10^{-8}$
14.2289	1	.995795	.995794	$-11\text{x}10^{-7}$
	2	-.0916176	-.0916177	$-90\text{x}10^{-9}$
	3	$.100723\text{x}10^{-4}$	$.100723\text{x}10^{-4}$	$27\text{x}10^{-13}$
	4	$.151197\text{x}10^{7}$	$.151197\text{x}10^{7}$	$15\text{x}10^{-1}$
	5	2.72320	2.72320	$20\text{x}10^{-8}$
15.7080	1	$.797718\text{x}10^{-6}$	$.587949\text{x}10^{-6}$	$-21\text{x}10^{-8}$
	2	-1.00000	-1.00000	$12\text{x}10^{-7}$
	3	$.251792\text{x}10^{-5}$	$.251792\text{x}10^{-5}$	$87\text{x}10^{-14}$
	4	$.663563\text{x}10^{7}$	$.663564\text{x}10^{7}$	$75\text{x}10^{-1}$
	5	2.81589	2.81589	$20\text{x}10^{-8}$

APPENDIX A
LISTING OF EXAMPLE NO. 1 WITH CONTROL CARDS FOR THE NOLOS SYSTEM USED AT NOL

```
CCAFNOL,T100,CM60000.SYSTEM,039,ORLOW.
ATTACH(ABC,NOLBIN)
COPYN(0,DEF,ABC)     FIRST PARAMETER IS THE NUMBER '0'
RETURN(ABC)
FTN(L)
LOAD(LGO)
DFF.
'      RECORD SEPARATOR = (7-8-9) PUNCH IN COL. 1
REWIND(ABC)
FNOL3,1,ABC
'      RECORD SEPARATOR = (7-8-9) PUNCH IN COL. 1
      PROGRAM EX1 (INPUT,OUTPUT)
C
C      EXAMPLE NO. 1
C      SOLVE THE INITIAL VALUE PROBLEM DY/DX=A*W*COS(W*X)
C      USING THE FOLLOWING METHODS
C
C      (I)   4TH ORDER RUNGE-KUTTA (CONSTANT STEP)
C      (II)  4TH ORDER ADAMS-MOULTON METHOD (VARIABLE STEP)
C      (III) 4TH ORDER RUNGE-KUTTA (VARIABLE STEP)
C
      DIMENSION Y(20),D(20)
      EXTERNAL DERIV,TERM,OUTPUT
      COMMON A,W
      A=1.
      W=1.
      DO 200 J=1,3
C         NUMBER OF EQUATIONS
      N=1
C         INITIAL STEP SIZE
      G=.1
C         NUMBER OF EXTRA Y'S
      L=2
C         NUMBER OF ACCEPTED STEPS TAKEN BETWEEN PRINTS
      M=5
C         ERROR WINDOW EXPONENT
      XNE=5.
      Y(1)=0.
      Y(N+3)=.01
      X=0.
      PRINT 1000
 1000 FORMAT (1H1,4X,1HX,17X,9HYT=TRUE Y,12X,13HYC=COMPUTED Y,11X,
     1        10H   (YT-YC))
      CALL FNOL3 (J,N,G,L,M,XNE,X,Y,D,DERIV,TERM,OUTPUT)
  200 CONTINUE
      STOP
      END
```

APPENDIX A CONTINUED

```
      SUBROUTINE DERIV (X,Y,D)
      DIMENSION D(1)
      COMMON A,W
      D(1)=A*W*COS(W*X)
      RETURN
      END
      SUBROUTINE TERM (X,Y,D,T)
      DIMENSION Y(1)
      COMMON A,W
      T=X-9.424778
      Y(5)=A*SIN(W*X)
      Y(6)=Y(5)-Y(1)
      RETURN
      END
      SUBROUTINE OUTPUT (X,Y,D,ERROR,N,L,H)
      DIMENSION Y(1)
      PRINT 1000, X,Y(5),Y(1),Y(6)
 1000 FORMAT (1H ,F10.6,10X,3(E13.6,10X))
      RETURN
      END
    ORLOW    039    CCAFNOL      END OF FILE = (6-7-8-9) PUNCH IN COL. 1
```

APPENDIX B
LISTING OF EXAMPLE NO. 2

```
      PROGRAM EX2(INPUT,OUTPUT)
C      EXAMPLE NO. 2
C
C      SOLVE D**2Y/DX**2=-Y
C
C      USING ADAMS-MOULTON PREDICTOR CORRECTOR WITH
C      STEP SIZE CONTROLLED BY
C
C      (I)   RELATIVE ERROR
C      (II)  ABSOLUTE ERROR
C
      DIMENSION Y(30),D(30)
      EXTERNAL DERIV,TERM,OUTPUT
C        ADAMS-MOULTON METHOD
      J=2
C        NUMBER OF EQUATIONS
      N=2
C        MINIMUM RELATIVE ERROR DIVISOR
      Y(N+3)=.001
C        NUMBER OF EXTRA Y'S
      L=2
C        NUMBER OF ACCEPTED STEPS BETWEEN PRINT CYCLES
      M=10
C        EXPONENT ERROR WINDOW
      XNE=5.
C        INITIAL STEP SIZE
      G=.01
C        INITIAL CONDITIONS
      X=0.
      Y(1)=0.
      Y(2)=1.
      PRINT 1000
 1000 FORMAT(1H1,4X,1HX,13X,13HYC=COMPUTED Y,16X,9HYT=TRUE Y,11X,
     1       10H   (YT-YC))
      CALL FNOL3(J,N,G,L,M,XNE,X,Y,D,DERIV,TERM,OUTPUT)
C        RESET THE PRINTING OPTION
      M=0
      Y(N+2)=1.
C        RESET THE INITIAL CONDITIONS
      X=0.
      Y(1)=0.
      Y(2)=1.
C        CHANGE TO ABSOLUTE ERROR
      Y(N+3)=-Y(N+3)
      PRINT 1000
      CALL FNOL3(J,N,G,L,M,XNE,X,Y,D,DERIV,TERM,OUTPUT)
      STOP
      END
```

APPENDIX B CONTINUED

```
      SUBROUTINE DERIV(X,Y,D)
      DIMENSION Y(1),D(1)
      D(1)=Y(2)
      D(2)=-Y(1)
      RETURN
      END
      SUBROUTINE TERM(X,Y,D,T)
      DIMENSION Y(1)
      T=15.707963-X
      Y(3)=X
      Y(6)=SIN(X)
      Y(7)=Y(6)-Y(1)
      RETURN
      END
      SUBROUTINE OUTPUT(X,Y,D,ERROR,N,L,H)
      DIMENSION Y(1)
      LF=N+4
      LL=N+3+L
      PRINT 1000, X,Y(1),(Y(I),I=LF,LL)
 1000 FORMAT(1H ,F10.6,10X,3(E13.6,10X))
      RETURN
      END
```

APPENDIX C
LISTING OF EXAMPLE NO. 3

```
      PROGRAM EX3(INPUT,OUTPUT)
C      EXAMPLE NO. 3
C
C      SOLVE THE SYSTEM OF FIRST ORDER EQUATIONS
C
C          DY(1)/DX=Y(2)
C          DY(2)/DX=-Y(1)
C          DY(3)/DX=(1./(X+1.)-1.)*Y(3)
C          DY(4)/DX=Y(4)-X+1.
C          DY(5)/DX=1./(X+1.)
C
C      USING ADAMS-MOULTON PREDICTOR CORRECTOR METHOD WITH
C      STEP SIZE CONTROLLED BY RELATIVE ERROR
C
      DIMENSION Y(30),D(30)
      EXTERNAL DERIV,TERM,OUTPUT
C         ADAMS-MOULTON METHOD
      J=2
C         NUMBER OF EQUATIONS
      N=5
C         MINIMUM RELATIVE ERROR DIVISOR
      Y(N+3)=.01
C         NUMBER OF EXTRA Y'S
      L=10
C         NUMBER OF ACCEPTED STEPS BETWEEN PRINT CYCLES
      M=40
C         EXPONENT ERROR WINDOW
      XNE=5.
C         INITIAL STEP SIZE
      G=.02
C         INITIAL CONDITIONS
      X=0.
      Y(1)=0.
      Y(2)=1.
      Y(3)=1.
      Y(4)=1.
      Y(5)=0.
      PRINT 1000
 1000 FORMAT(1H1,4X,1HI,13X,14HYC=COMPUTED YI,14X,10HYT=TRUE YI,11X,
     1        10H   (YT-YC))
      CALL FNOL3(J,N,G,L,M,XNE,X,Y,D,DERIV,TERM,OUTPUT)
      STOP
      END
```

APPENDIX C CONTINUED

```
      SUBROUTINE DERIV(X,Y,D)
      DIMENSION Y(1),D(1)
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=(1./(X+1.)-1.)*Y(3)
      D(4)=Y(4)-X+1.
      D(5)=1./(X+1.)
      RETURN
      END
      SUBROUTINE TERM(X,Y,D,T)
      DIMENSION Y(1)
      T=X-5.*3.141592536
      Y(9)=SIN(X)
      Y(10)=COS(X)
      Y(11)=(X+1.)*EXP(-X)
      Y(12)=X+EXP(X)
      Y(13)=ALOG(X+1.)
      DO 10 I=14,18
      Y(I)=Y(I-5)-Y(I-13)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE OUTPUT(X,Y,D,ERROR,N,L,H)
      DIMENSION Y(1)
      PRINT 1000, X
 1000 FORMAT(1H0,50X,3HX= ,F7.4)
      J=N+3
      K=2*N+3
      DO 20 I=1,N
      J=J+1
      K=K+1
      PRINT 2000, I,Y(I),Y(J),Y(K)
 2000 FORMAT(1H ,I5,15X,3(E13.6,10X))
   20 CONTINUE
      RETURN
      END
```

APPENDIX D
LISTING OF FNOL3

```
      SUBROUTINE FNOL3(J,NN,G,L,MPR,XNE,X,Y,D,DERIV,TERM,OUTPUT)
C 001 J=INTEGRATION METHOD CONTROL
C 002 NN=NUMBER OF SIMULTANEOUS DIFFERENTIAL EQUATIONS
C 003 G=FIRST INTERVAL OF INTEGRATION
C 004 L=NUMBER OF Y$S GREATER THAN N TO BE PRINTED
C 005 MPR=PRINT FREQUENCY
C 006 XNE=CONTROL FOR INTERVAL OF INTEGRATION
C 007 X=INDEPENDENT VARIABLE
C 008 Y=DEPENDENT VARIABLE
C 009 D=ARRAY CONTAINING DERIVATIVES
C 010 DERIV=SUBROUTINE IN WHICH DIFFERENTIAL EQUATIONS ARE FOUND
C 011 TERM=SUBROUTINE FOR TERMINATION CONDITION
C 012 OUTPUT=SUBROUTINE FOR PRINTING OUTPUT
C     NRET=TERM LOOP COUNTER
C     NPR=COUNT OF STEPS SINCE LAST PRINT
C     PC=Y(N+1)=PRINT CONTROL OTHER THAN STEP COUNT
C     JJ=J-2=0 FOR AM,-1 FOR RK,+1 FOR RK2
C     MK=AM RK STEP COUNT
C     XD=DP FORM OF X
C THIS SUBROUTINE CAN ALSO BE RUN IN DOUBLE PRECISION BY REMOVING
C THE 'C' IN COLUMN 1 ON THE DOUBLE PRECISION STATEMENT
C     DOUBLE PRECISION XD,YD,YP,YC,YK,H,HC,XS,XDS,HS,HN,HD24,HD6
      DIMENSION C(3),Y(30),YD(30),YP(30),YC(30),D(50),DM(30,4),DK(30,4)
     1,ERROR(30),YK(30)
      DATA EP6,EP11,M4/1.E-6,1.E-11,-4/
      DATA (C(K),K=1,3)/2*.5,1./
      DATA HMAX5/1.E35/
      NHTS=0
      FP2=0.
      N=NN
      JJ=J-2
      H=G
      HN=H
      MK=1
      NRET=M4
      JTEST=1
      IF (JJ .LT. 0) JTEST=4
      IF (XNE.EQ. 0.) GO TO 15
      EC=Y(N+3)
      EUP=10.**(-XNE)
      ELO=EUP*.001
      EM=ELO*31.6227766
   15 XD=X
      XS=XD
      DO 20 I=1,N
      ERROR(I)=0.
   20 YD(I)=Y(I)
      CALL DERIV(X,Y,D)
      CALL TERM (X,Y,D,F)
C     PRINT
```

```
   50 CALL OUTPUT(X,Y,D,ERROR,N,L,H)
      IF (NRET) 65,60,55
   55 PRINT 3000, HN
 3000 FORMAT(108H1EXECUTION TERMINATED BECAUSE INTERVAL OF INTEGRATION L
     1ESS THAN 1.0E -6 TIMES INDEPENDENT VARIABLE (X).  H =,1PE15.7)
      STOP
   60 RETURN
   65 NPR=0
      IF (MPR .LE. 0) PC=Y(N+1)
  100 IF (JTEST .EQ. 5 .AND. H .NE. HN) GO TO 455
      IF (JJ .GE. 0) H=HN
      IF (MK .NE. 0 .OR. JJ .NE. 0) GO TO 300
C------THE ADAMS MOULTON METHOD
  200 HD24=H/24.
      JAM=0
      DO 210 I=1,N
      YPI=(55.*DM(I,1)+37.*DM(I,2))-(59.*DM(I,3)+9.*DM(I,4))
      YP(I)=YD(I)+HD24*YPI
      Y(I)=YP(I)
  210 CONTINUE
      X=XD+H
      CALL DERIV(X,Y,DM(1,4))
      DO 220 I=1,N
      YPI=(9.*DM(I,4)+19.*DM(I,1)+DM(I,2))-5.*DM(I,3)
      YC(I)=YD(I)+HD24*YPI
      ERROR(I)=(YP(I)-YC(I))/14.
C THIS ADDS IN A 2D CORRECTION
      YC(I)=YC(I)+ERROR(I)
  220 CONTINUE
      IF (XNE.NE..0) GO TO 700
      GO TO 455
C------THE RUNGE KUTTA METHOD
  300 GO TO (301,309,308,309,303),JTEST
  301 DO 302 I=1,N
      YK(I)=YD(I)
  302 CONTINUE
      XDS=XD
      GO TO 309
  303 DO 304 I=1,N
      YK(I)=YC(I)
  304 CONTINUE
      XDS=XD+H
  308 HS=H
      H=2.*H
      GO TO 320
  309 X=XD
      JAM=1
      DO 310 I=1,N
      Y(I)=YD(I)
      DK(I,1)=D(I)
  310 CONTINUE
```

```
      IF (JTEST .LE. 2) CALL DERIV(X,Y,DK)
      IF (MK .GT. 1 .OR. JTEST .GT. 1) GO TO 320
      DO 315 I=1,N
      DM(I,4)=DK(I,1)
  315 CONTINUE
  320 DO 335 K=2,4
      HC=H*C(K-1)
      DO 330 I=1,N
      Y(I)=YD(I) + HC*DK(I,K-1)
  330 CONTINUE
      X=XD+HC
      CALL DERIV(X,Y,DK(1,K))
  335 CONTINUE
      HD6=H/6.
      DO 340 I=1,N
      YPI=DK(I,1)+DK(I,4)+2.*(DK(I,2)+DK(I,3))
      YC(I)=YD(I)+HD6*YPI
  340 CONTINUE
      GO TO (360,390,370,455,370),JTEST
  360 DO 365 I=1,N
      YP(I)=YC(I)
  365 CONTINUE
      JTEST=3
      GO TO 308
  370 DO 380 I=1,N
      YD(I)=YP(I)
      YP(I)=YC(I)
  380 CONTINUE
      H=HS
      XD=XD+H
      JTEST=2
      IF (MK .EQ. 1) GO TO 309
      GO TO 451
  390 DO 400 I=1,N
      ERROR(I)=(YC(I)-YP(I))/15.
      YC(I)=YC(I)+ERROR(I)
      YP(I)=YC(I)
  400 CONTINUE
      JTEST=5
      IF (XNE.NE..0) GO TO 700
C------ACCEPT THE STEP SIZE
  450 IF (JAM .EQ. 0) GO TO 455
      IF (MK .EQ. 3 .AND. JJ .EQ. 0) GO TO 455
      IF (MK .NE. 1) GO TO 303
      IF (JTEST .EQ. 1) GO TO 455
  451 DO 452 I=1,N
      Y(I)=YD(I)
  452 CONTINUE
      GO TO 466
  455 DO 459 NQ=1,N
      YD(NQ)=YC(NQ)
```

APPENDIX D CONTINUED

```
      Y(NQ)=YD(NQ)
  459 CONTINUE
      IF (JJ .GE. 0) JTEST=1
      IF (MK .NE. 0 .OR. JJ .NE. 0 .OR. NRET .NE. M4) GO TO 465
      DO 460 I=1,N
      DM(I,4)=DM(I,2)
      DM(I,2)=DM(I,3)
      DM(I,3)=DM(I,1)
  460 CONTINUE
  465 XD=XD+H
  466 X=XD
      IF (MK .EQ. 3) MK=0
      CALL DERIV(X,Y,D)
      DO 470 I=1,N
      DM(I,MK+1)=D(I)
  470 CONTINUE
      FP=F
      CALL TERM (X,Y,D,F)
C------DO YOU TERMINATE
  500 IF (ABS(F) .LE. EP6) GO TO 800
      IF (FP .EQ. 0.) GO TO 510
      IF (NRET .NE. M4 .OR. F*FP .LT. EP11) GO TO 805
  510 XS=XD
      IF (MK .NE. 0 .AND. H .EQ. HN) MK=MK+1
C------DO YOU PRINT
  600 NPR=NPR+1
      IF (MPR .LE. 0) GO TO 610
      IF (NPR .GE. MPR) GO TO 50
      GO TO 100
  610 IF (ABS(Y(N+1)-PC) .GE. Y(N+2)) GO TO 50
      GO TO 100
C------DETERMINING THE STEP SIZE
  700 HB = HMAX5
      DO 760 I = 1,N
      Z=ABS(ERROR(I))
      IF (YC(I) .EQ. 0.) GO TO 720
      ZZ=YC(I)
      ZZ=ABS(ZZ)
      IF (EC) 720,710,705
  705 IF (EC .GT. ZZ) ZZ=EC
  710 Z=Z/ZZ
  720 IF(Z.GT.ELO.AND.Z.LT.EUP) GOTO 750
      HB = AMIN1(HB,EM/(Z+EP11))
      GOTO760
  750 HB=AMIN1(HB,1.)
  760 CONTINUE
      IF (HB .NE. 1.) GO TO 765
      NHTS=0
      GO TO 450
  765 HN=H*HB**.2
      IF (MK .NE. 1) JTEST=1
```

```
      MK=1
      IF (HB.LT.1.) GOTO 770
      IF (ABS(HN) .GT. ABS(4.*H)) HN=4.*H
      NHTS=0
      GOTO 450
  770 HEPS=ABS(X*EP6) + EP11
      IF (ABS(HN) .LT. ABS(H/4.)) HN=H/4.
      IF (ABS(HN) .GT. HEPS) GO TO 790
      NHTS = NHTS + 1
      IF (NHTS .LE. 10) GO TO 780
      NRET = 1
      GO TO 450
  780 HN=SIGN(HEPS,HN)
      IF (NHTS .GT. 1) GO TO 450
  790 IF (NHTS .GT. 1) NHTS=0
      IF (JAM .EQ. 0) GO TO 100
      DO 795 I=1,N
      YD(I)=YK(I)
  795 CONTINUE
      XD=XDS
      JTEST=1
      GO TO 100
C------THE TERMINATION LOOP
  800 NRET=0
  805 IF (NRET .LT. 0) GO TO 806
      H=XD-XS
      GO TO 50
  806 IF(F*FP.LT.0.) GOTO 810
      IF (F*FP2.LT.0.) GOTO 820
      GO TO 800
  810 FP2 =FP
      HP =H
      GOTO 830
  820 FP =FP2
      HP =H + HP
  830 NRET=NRET+1
      H=HP*F/(FP-F)
      JTEST=4
      GOTO 300
       END
```

APPENDIX E

FNOL3 VS. FNOL2

E-1. FNOL3 does not allow the step size to be changed by more than a factor of four. In FNOL2, the step size is allowed to change by any factor, which can cause the step size to be too large or too small in some cases.

E-2. If the step size is too small, FNOL3 accepts the step and the step size is increased for the next step. FNOL2 does not accept the step.

E-3. FNOL3 calls the output routine only if the step was accepted. FNOL2 calls the output routine and then checks to see if the step is acceptable.

E-4. In FNOL3, the truncation errors are added at each step to the dependent variables. FNOL2 does not add the truncation errors to the dependent variables.

E-5. FNOL3 allows the step size to be changed when using the Runge-Kutta method. The step size cannot be changed using the Runge-Kutta method in FNOL2.

E-6. In FNOL3, the option was added that if relative error is used to adjust the step size and the user sets $Y(N+3)=\varepsilon (\varepsilon > 0)$, then whenever $|Y(I)| < Y(N+3)$, relative error is calculated by dividing truncation error by Y(N+3) rather than Y(I). Otherwise perfect accuracy would be required of every variable passing through 0.

E-7. FNOL3 uses XNE, a floating point rather than an integer, to allow more control over the step size.

APPENDIX F

FNOL3 ARGUMENT LIST SUMMARY

ARGUMENT	TYPE	INPUT	OUTPUT	REMARKS
J	INTEGER	YES	____	The integration method
N	INTEGER	YES	____	The number of equations
G	REAL	YES	____	The initial step size
L	INTEGER	YES	____	The additional output Y's
M	INTEGER	YES	____	The print frequency
XNE	REAL	YES	____	The control for step size
X	REAL	YES	YES	The independent variable
Y	REAL	YES	YES	The dependent variables
D	REAL	NO	YES	The derivatives
DERIV	____	____	____	CALL DERIV(X,Y,D)
TERM	____	____	____	CALL TERM(X,Y,D,T)
OUTPUT	____	____	____	CALL OUTPUT(X,Y,D,ERROR,N,L,H)

MATHEMATICS DEPARTMENT
EXTERNAL DISTRIBUTION LIST

	Copies
Commander, Naval Air Systems Command Department of the Navy Washington, D.C. 20360 Attn: Code 604 Attn: Code 03D Attn: Code 5203	
Commander, Naval Ordnance Systems Command Department of the Navy Washington, D.C. 20360	
Attn: Code 9132	2
Attn: Code 03A Attn: Code 0322 Attn: Code 083213 (Boucher)	
Director, Special Projects Department of the Navy Washington, D.C. 20360 Attn: SP-2722	
Commander, Naval Ship Systems Command Department of the Navy Washington, D.C. 20360 Attn: Code 031 (J. H. Huth)	
Chief, Defense Atomic Support Agnecy Washington, D.C. 20390	
Attn: Document Library Branch	2
Commanding General Aberdeen Proving Ground, Maryland 21005	
Attn: Technical Information Branch	2
Attn: Director, Ballistic Research Laboratories	2
Defense Documentation Center Cameron Station	
Alexandria, Virginia 22314	20
Commanding Officer and Director Aerodynamics Laboratory Naval Ship Research and Development Center Washington, D.C. 20007 Attn: Library	

Copies

Commanding Officer and Director
Applied Mathematics Laboratory
Naval Ship Research and Development Center
Washington, D.C. 20007
Attn: Dr. D. Shanks
Attn: Library
Attn: J. A. Blalock

Commanding Officer
Harry Diamond Laboratory
Washington, D.C. 20438
Attn: Library, Bldg. 92, Rm. 211

Director of Intelligence
Headquarters, USAF
Washington, D.C. 20390
Attn: AFOIN-3B

NASA
Langley Research Center
Langley Field, Virginia 23365
Attn: Theoretical Aerodynamic Division
Attn: Librarian

NASA
Ames Research Center
Moffett Field, California 94035
Attn: Librarian

NASA
High Speed Flight Station
Edwards, California 93523
Attn: Library

NASA
Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44134
Attn: Librarian
Attn: Chief, Propulsion and Aerodynamic Division

Commander
Pacific Missile Range
Point Mugu, California 93041
Attn: Technical Library
Attn: C. J. Thorne
Attn: J. C. Alexander

National Bureau of Standards
Washington, D.C. 20390
Attn: Applied Mathematics Division
Attn: Library

Copies

Commander
U. S. Naval Ordnance Test Station
China Lake, California 93555
Attn: Technical Library
Attn: Code 406
Attn: Code 507
Attn: Code 5019
Attn: A. A. Fojt

Commander
U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448
Attn: Technical Library
Attn: E. C. Hubbard

Director
Naval Research Laboratory
Code 2029 (ONRL)
Washington, D.C. 20390
Attn: Library

Office of Naval Research
800 N. Quincy Street
Arlington, Virginia 22217
Attn: Head, Mathematics Program
Attn: Dr. Stuart L. Brodsky, Code 432

Naval Research Laboratory
Washington, D.C. 20390
Attn: Technical Documents
Attn: Code 2027

Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830
Attn: Dr. A. S. Householder
Attn: Library

Commander
Naval Ordnance Laboratory
Corona, California 91720
Attn: A. Sevy

Commander
Naval Missile and Astronautics Center
Point Mugu, California 93041
Attn: C. A. Kent

Office of Assistant Secretary of Defense (R&D)
Room 3E1065, The Pentagon
Washington, D.C. 20390
Attn: Technical Library

Copies

Headquarters, Arnold Engineering Development Center
Air Research and Development Center
Arnold Air Force Station, Tennessee 37389
Attn: Technical Library

Commanding General
Redstone Arsenal
Huntsville, Alabama 35806
Attn: Technical Library

Hughes Aircraft Company
Florence Avenue at Teale Street
Culver City, California 90230
Attn: Mr. Dana H. Hohnson
R&D Technical Library

McDonnell Aircraft Corporation
P.O. Box 516
St. Louis, Missouri 63166
Attn: Engineering Library
Department 644

United Aircraft Corporation
400 Main Street
East Hartford, Connecticut 06118
Attn: Chief Librarian

Superintendent
U. S. Naval Postgraduate School
Monterey, California 93940
Attn: Library, Technical Reports Section

California Institute of Technology
Pasadena, California 91109
Attn: Aeronautics Department

Aero Engineering Department
University of Maryland
College Park, Maryland 20740
Attn: Prof. A. Wiley Sherwood

Lockheed Missiles and Space Company
P.O. Box 504
Sunnyvale, California 94088
Attn: Technical Information Center

Guggenheim Aeronautical Laboratory
California Institute of Technology
Pasadena, California 91109
Attn: Aeronautics Library

Copies

University of California
Berkeley, California 94720
Attn: Dr. S. A. Schaaf

University of California
Los Angeles, California 90024
Attn: Prof. M. R. Hestenes

California Institute of Technology
Pasadena, California 91109
Attn: Library

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109
Attn: Library
Dr. E. C. Posner

Mathematics Research Center
U. S. Army
University of Wisconsin
Madison, Wisconsin 53706
Attn: S. J. Kleene
E. Zarontonello

Case Institute of Technology
Cleveland, Ohio 10900
Attn: Prof. G. Kuerti

Cornell University
Graduate School of Aeronautical Engineering
Ithaca, New York 14850
Attn: Prof. W. R. Sears

Applied Physics Laboratory
Johns Hopkins University
8621 Georgia Avenue
Silver Spring, Maryland 20910
Attn: Dr. N. H. Choksy
C. V. Bitterli
Dr. S. Haywood

Chairman
Engineering and Applied Sciences
California Institute of Technology
Pasadena, California 91109

Mathematics Department
Harvard University
Cambridge, Massachusetts 02138
Attn: Prof. G. Birkhoff

Copies

Institute for Fluid Dynamics and Applied Mathematics
University of Maryland
College Park, Maryland 20740
Attn: Prof. M. H. Martin

University of Michigan
Randall Laboratory
Ann Arbor, Michigan 48106
Attn: Prof. Otto Laporte

Supervisor
Technical Documents Services
Willow Run Laboratories
University of Michigan
Willow Run Airport
Ypsilanti, Michigan 48197

Applied Mathematics and Statistics Laboratory
Stanford University
Stanford, California 94305
Attn: Prof. G. E. Forsythe

Defense Research Laboratory
The University of Texas
P.O. Box 8092
Austin, Texas 78767
Attn: H. D. Kirck, Asst. Director

The Avco Manufacturing Corporation
Research Laboratories
2385 Revere Beach Parkway
Everett, Massachusetts 02149

Cornell Aeronautical Laboratory Incorporated
P.O. Box 235, 4455 Genessee Street
Buffalo, New York 14225
Attn: Librarian

Douglas Aircraft Company Incorporated
Santa Monica Division
3000 Ocean Park Blvd.
Santa Monica, California 90405
Attn: Library

Missile and Ordnance Systems Department
General Electric Company
3198 Chestnut Street
Philadelphia, Pennsylvania 19104
Attn: L. Chasen, Mgr. Library

Goddard Space Center
Code 507
Greenbelt, Maryland 20770
Attn: Mr. Charles Newman
Mr. Jerry Linnekin

Smithsonian Institution
10th and Jefferson Drive
Washington, D.C. 20560
Attn: Mr. James Crockett
Attn: Mr. Dante Piacesi

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Ordnance Laboratory White Oak, Silver Spring, Maryland 20910	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

FNOL3, A Computer Program to Solve Ordinary Differential Equations

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Ralph E. Ferguson
Theodore A. Orlow

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
1 March 1971	40	2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) NOLTR 71-2
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale, its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

FNOL3 is a Fortran IV subprogram which uses fourth order Runge-Kutta and Adams-Moulton methods to solve up to 30 ordinary differential equations with initial conditions. Options allow control of error, step-size, print frequency and integration method. There is a discussion on the integration methods and their error terms. Listings of the subprogram and sample problems and their results as run on NOL's CDC 6400 are included.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
FNOL3						
Ordinary Differential Equations						
Runge-Kutta						
Adams-Moulton						
CDC 6400						
Numerical Integration						
Initial Conditions						
Fortran IV						
Computer Digital						

DD FORM 1 NOV 65 1473 (BACK)

(PAGE 2)

Unclassified

Security Classification

NOL LIBRARY REPORT NUMBER (LIST ONE REPORT ONLY)			DATE REQUESTED
NOLTR 71-2			12/12/75
NAME	DIVISION	ROOM NO.	EXT.
Jerry M. Ward	WR-15	336-210	4-1169

STATUS OF REQUEST *(FOR 730 USE ONLY)*

☐ ON LOAN WITHIN NOL AND BEING RECALLED

☐ ON MICROFILM - MICROFILM AND READER IN ROOM 1-328

☐ YOUR NAME ADDED TO WAITING LIST

☐ REPORTS DESTROYED IN ACCORDANCE WITH NOLINST. 5070.1B

☐ RECALLED FROM STORAGE AVAILABLE ______ (DAYS) ______ (WEEKS)

☐ OTHER

REPORT REQUEST NDW-NOL 5070/21 (REV. 8-66)
WHITE COPY-FOR LIBRARY USE   YELLOW COPY-FOR REQUESTER'S USE